Oifig Na
bPaitinní

The
Patents
Office

IE 02/0429

ABSTRACT

"SYSTEM METHOD FOR IDENTIFICATION AND AUTHENTICATION OF INFORMATION PROCESSING DEVICES"

A system to obtain unique fingerprints from computer equipment is presented. The system is able to probabilistically discriminate between two computers with an arbitrary degree of certainty, The Fingerprint of a system is obtained as a combination of information that is unique to the hardware and information about its configuration and state. <Fig.4>

SYSTEM AND METHOD FOR IDENTIFICATION AND AUTHENTICATION OF INFORMATION PROCESSING DEVICES

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of digital devices and systems. More particularly, the present invention relates to identifying such digital devices and systems.

2. Discussion of Related Art

At present no universally accepted method exists for developing measurements which uniquely identify a digital device or system based on their physical characteristics. Such an identification method is highly desirable for authenticating remote access providers. Copyright infringement could be prevented by authenticating the system on which music is being played, videos are being displayed, and software is being executed using a unique identifier based on the physical characteristics of the system. Any system providing use on a restricted basis can benefit from the security provided by unique identifiers based on physical device properties.

The prior art fails to provide a unique identifier that is immune to tampering.

SUMMARY OF THE INVENTION

A unique and private identifier that provides discrimination between two digital devices exemplified hereinafter by computers (and in general, two electronic appliances) is important for ensuring security and accountability in many applications. For cryptographic applications, the availability of a computer fingerprint that cannot be faked or duplicated by an attacker can be used to set a Certificate Authority Scheme such as the one shown schematically in Figure 1.

Typically, computing devices are identified by hardware serial numbers or software files called keys. Both are easily obtainable by third parties and can be used against the legal owner of the information. Publication and use of hardware serial numbers are also considered by many as a breach of personal freedom. Keys are software files which can be easily stolen, thus placing a tremendous responsibility on the owner of the keys. Mismanagement of keys have breached many security and copy protection systems. Most notably, CSS, the copy protection scheme on DVD movies, was broken one year after its public debut resulting in a loss of copy protection for the remaining lifecycle of DVD movies and multi-billion dollar losses to the movie industry. Both serial numbers and keys can be metaphorically considered as passports, driver's licenses or birth certificates to humans. However when we need to identify someone with a higher degree of confidence one must resort to biometric techniques. It is easier to falsify the documents identifying a person than their physical characteristics.

The present invention provides a methodology for identification that possesses an arbitrary degree of confidence. The method develops a fingerprint based on measurements of analog artifacts exposed during processing by a particular computing device. In the context of the present invention, a computing device refers to a single processing unit or to several processing units interconnected to form a network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a certificate authority scheme using a prior art fingerprint technology.

FIG. 2 illustrates a method of analysis consisting of measuring the deviations from a linear regression model of the data obtained in a first set of m tests.

FIG. 3 illustrates a flow chart of development of a CMOS fingerprint.

Fig. 4 illustrates a network fingerprinting scheme.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENT

A minimal model of a 'device' that is to be identified is one composed of at least a processing unit (CPU), a memory unit (MEM comprising RAM and ROM) and a clock (CLK) that sets the pace and synchronizes the operation of the component parts. Most practical devices will have additional storage devices (disk, tapes), communications devices (network cards, modems) and interfacing devices controllers (video, keyboards, mouse, etc). Software, communication protocols and processes can be considered as integrating parts of the system for identification purposes. Given the enormous variety of designs of computer systems, this classification is only descriptive of the functionality of the components, and is not provided as limiting in any sense.

The most intrinsic effect of the physical layout of the components is a consequence of the finite speed at which information propagates inside a computer. The absolute limit at which the electrical impulses can travel is given by the (finite) speed of light (~2.99 x 10^9 m/sec). Table 1 gives the order of magnitude of the times required for electrical impulses to clear some typical distances found in modem computing hardware.

From the measurements of propagation times in a computer hardware system, one can conclude that the physical layout (relative distance between components) of a particular circuit has an influence on the response time of the corresponding device. Therefore, to discriminate between two different layouts a very refined clock is needed. However, in typical digital devices, it takes many information exchanges, whose number depends on the particular hardware and software being used, to access or process a particular piece of information. The combination of finite speed of propagation for the information and the necessary synchronization operations between diverse components, gives rise to random delays in response times called latency. This synchronization is often regulated by a common clock signal carried by the control bus. A conceptual connection between latency and entropy in a physical system can be made in the sense that latency is a measure of the degree of uncertainty about the state of the system.

In addition, in the manufacturing process of any device, there are tolerable imperfections introduced. These are differences that do not compromise the functionality of the device so long as component performance lies within certain bounds. For the purposes of the present invention, these imperfections are a way to characterize and distinguish a particular component from other components made in the same production line at an equivalent time (same processes, same equipment, same state for the production line). In principle, no two components possess exactly the same tolerable imperfections, therefore they should not respond in exactly the same way to the same request. However, once a response is established, e.g. propagation time, the response must be consistent, at least in a statistical sense, from trial to trial in order to be usable as an identifier.

Differences in architecture result in systems providing different responses to the same stimuli, the response being statistically the same for the same machine and different for different machines. It is possible, in principle, to differentiate between systems through the analysis of their individual responses to identical stimuli.

Distance	Typical length	Characteristic Time
Transistors inside a chip	1 cm	3 x 10^{-15} sec
Across the chip	1 cm	3 x 10^{-11} sec
Across the Motherboard	10 cm	3 x 10^{-10} sec

Table 1: Time taken by a signal traveling at light-speed inside a computer

Strong candidates for use as such stimuli are read/write operations. Read/write operations are controlled by a memory controller which performs logical to physical translation processes tailored to each storage device. These controller processes comprise algorithms employing tables and directories in order to map between the device's physical storage addresses and logical addresses. The relatively long sequence of operations needed to store/retrieve a byte is slow, when compared to the electronic transit times. The average difference between electronic transit time and the time to store/retrieve a byte is the average latency of the storage device.

As a non-limiting example, for solid state or dynamic random access memory (DRAM), the column access strobe (CAS) rating is used to describe how many clock cycles are required for a particular bit of information to be retrieved from a given storage device. CAS latency refers to the number of clock cycles it takes before a column can be addressed on a DRAM chip. Latency is a measure of delay, so a 'CL*n*' CAS latency factor indicates an *n*-clock cycle delay.

A different set of physical rules applies to mechanical storage devices such as hard drives. In this case, the average retrieval time is related to the rotation speed of the device, the deviations from the average being a consequence of the dynamic characteristics of the device. In particular, the influence of turbulence has been documented as a source of uncertainty or latency. In the case of disks, the relatively long response times of the mechanical components are a determining factor in information retrieval time. However, the same general considerations concerning the function of the controller unit can be made.

The general idea underlying the present invention can be stated simply as: Given a minimal appliance consisting of CPU + MEM + CLK, the latency and the tolerable imperfections in the components and in the assembly of the system together determine a particular probability distribution for each of the random variables governing the response times for a set of measurements. Knowledge of these statistical distributions can be used to characterize or identify a particular physical computer system. In other words, the method of the present invention magnifies and uses, for the purposes of identification, the analog effects that are inherent in the physical performance of a system comprising a plurality of cooperating digital devices and components.

Another important source of implicit uniqueness is provided by the explicit intrinsic information concerning a system. This information is unique to each system, but is explicitly available to external entities (public) or to the operator/administrator of the system (private). For example, hardware type and serial numbers arc unique to each system but they are exposed explicitly to the operating system and the public and are, therefore, susceptible to being exploited.

It is possible to associate some commonly used elements of computer identification with the concept of ID cards or credentials given to people as means of identification or certification of identity. For example, the internet protocol (IP) address of a particular node or the computer name inside a network act as unique identifiers in the same sense that telephone number and name act as unique identifiers for persons. This type of identifier can be arbitrarily changed at any time. In general, this type of identifier is publicly exposed as a mean of identification. Because of this public exposure, this information cannot be deemed as unique to a particular system. Further, two or more computers can be given the same name.

There are some other pieces of information, such as the CPU serial number, that are unique to each system. These pieces of information can be more or less difficult to obtain from outside the system, but are always exposed explicitly to the operating system. Table 2 illustrates some examples of identity information that can be used in an authentication scheme. For identification purposes, uniqueness is easily derivable from a mixture of information that is public and unique, however for authentication we need to have at least some private and unique information. Unconditional authentication can only be achieved if the private information is not explicit. Computer metrics can fulfill this last requirement because collection of such metrics requires overt stimulation of the system.

Piece of Info	Intrinsic	Extrinsic	Public	Private	Unique
CPU Serial Number	Y			Explicit	Y
Network Card MAC Address		Y	Y		Y
Computer Metrics	Y			Implicit	Y
Plug ‘n Play Configuration		Y		Explicit	
Hardware List and Specs	Y			Explicit	
Software List and Rev. No.		Y		Explicit	
IP Address		Y	Y		

Table 2: Classification of some pieces of information available for identification of a computer system.

One way to acquire information that possesses both intrinsic and implicit uniqueness is to perform timed tests on the devices and identify unique patterns in the statistical distributions of the measurements obtained. On most systems, the finest possible clock available for such purposes is the same system clock that the CPU utilizes to synchronize its functions. For example, on Intel® Pentium® chips there is a special 64-bit registry call the Time Stamp Counter (TSC) that records clock cycles.

A set of intrinsically and implicitly unique information can be acquired by performing and timing a pre-arranged series of tasks. A preferred choice of tasks is one such that all different components of a system are tested with varying degrees of load. For example, the distribution of elapsed times for performing a combination of memory-swapping and processor intensive tasks, provides information concerning the architecture of a system.

The combined information gathered from measurements involving more components of a system increases the degrees of freedom of the timing distributions, making it easier to discriminate among systems. To achieve consistency, absolute control over the process to be measured must be maintained. For example, Windows 2000 on an Intel i386-based central processing unit (CPU) uses a distinct paging system for addressing virtual memory. Manufacturers employing this CPU claim, under normal operation, to have a 90% hit rate on the cache (90% of memory accesses result in addresses that are in pages in the cache and require no page swapping). Consistency of stimuli, e.g., the same sequence of memory accesses, is required to override the Windows 2000 page caching system so that there is certainty that exactly the same phenomenon is measured every time measurements are taken using a given stimuli.

To find the characteristic time bounds for a certain system may require the collection of a considerable number of data points, depending on the precision sought for the system identification, i.e., the more bits desired in a fingerprint for a system, the more points needed to achieve effective differentiation between systems.

A more efficient way to characterize a system employs information concerning the distribution of series of timing values, thereby reducing the quantity of trials needed to obtain a fingerprint of the required bit-length. A distribution can be characterized by its moments. It is standard to characterize probability distributions by their first and second moments (usually referred to as mean and variance respectively), but if more parameters are needed, higher order moments can be employed. Care shall be taken concerning the accuracy of these statistical values, i.e., the more points measured and included in these calculated moments, the lower the error in the calculation of these parameters. Therefore a minimum number of samples should be determined for the accuracy required. The number of samples measurements obtained is influenced by compromising between accuracy and speed. In a preferred embodiment, successive moments of the distributions of series of timing values to characterize a particular system will be calculated.

Other possibilities for data analysis include obtaining integral parameters or deviations from these. As a non-limiting example, in a set of n measurements of a variable, the average taken over the first m < n samples can be calculated. This average is then used to calculate the variance of the rest of the observations with respect to the values obtained by using a linear regression model, as illustrated in Figure 2.

The concept of fingerprinting can be extended to individualize an entire network of computers. Although statistical analysis of network traffic patterns has been extensively studied in the context of Intrusion Detection Systems for network administration, see M. Burgess, "Thermal Non-Equilibrium Phase Space for Networked Computers", Physics Review E, 62:1736, 2000, the treatment of data is different for fingerprinting purposes. In the case of intrusion detection, the state of a computer system is defined as a function of the time consumption for a known task and the assumption is made that the time required for the computing network to perform this task is within certain bounds that uniquely characterize the system. In the context of fingerprinting applications, it may be necessary to have to partially or completely halt normal traffic on the network in order to develop a fingerprint that is unique to the network.

As a further non-limiting example (and empirical proof of the concept), consider an experiment in which a simple code writes to the available 50 bytes of the CMOS in Intel Pentium chips of two nearly identical systems A and B. That is, A and B are two similar systems having the same architecture and components, running the same operating system, and with serial numbers indicating manufacture at essentially the same point in time (using the identical production line). In this example, the time taken to fill the 50 bytes for a fixed number of repetitions is logged from the TSC registry. A flow chart of this procedure is illustrated in FIG. 3. A fixed pattern of repetitions of the procedure results in a file with information about each repetition. Analysis of statistical parameters of logged series of measurements revealed that it is possible to employ these time measurements to discriminate between such nearly identical systems. That is, it is possible to distinguish, with error probability less than 1/2, whether the flipping of 50 bytes is taking place in a particular system. Moreover, when the systems being measured are running under different environmental conditions, changes in the logged write times occur.

In a personal computer, various other possibilities exist for development of time series for accesses to devices on the PCI bus (network cards, graphics cards, etc.) and IDE devices (hard drives, disk drives, CD-ROMS, etc.). Information obtained for these devices provide more variety and possibilities for obtaining a fingerprint of the system.

Thus, a unique identification for a system can be readily obtained and input to a fingerprint creation process. For device to device authentication, this explicit unique identity can be combined with intrinsic and private identity in a typical authentication scheme such as a hash based challenge-response or a zero knowledge proof system.

In a challenge-response system, System A sends a log-on request to System B and System B replies with a randomly generated "token" (or challenge). System A hashes the currently logged-on user's cryptographically protected password with the challenge and sends the resulting "response" to System B. System B receives the challenge-hashed response and compares it to what it knows to be the appropriate response. (System B takes a copy of the original token - which it generated - and hashes it against what it knows to be the user's password hash from its own database.) If the received response matches the expected response, System A is successfully authenticated to System B.

A zero-knowledge proof is a protocol that proves that a person or system does have a piece of information, but it does not give away the piece of information or any way of determining the piece of information.

To individualize a specific user, explicit and intrinsic private uniqueness can be combined with a user's password or passphrase for a hash-based challenge-response or zero knowledge system. The combination of the user's passphrase and the computer's identification will suffice to track and identify a particular user.

At a higher level in the computer other intrinsic uniqueness such as network location and data location can be employed. Network location can include routing information relative to other predetermined network locations such as average transmission and response times for these other locations. Data location can be measured in two ways. At a low level, read times can be measured for file locations on the hard disk that are not typically moved by disk defragmentation programs, and are repeatable. These files are typically system files first loaded onto the system during its initial installation. Conversely, read/write times can be measured for contents within a block on the hard drive in a location that is typically untouched by disk defragmentation programs, which is also repeatable.

At a still higher level, the user may wish to use a specific floppy disk and/or a CDROM to help identify the system. This approach has the disadvantage that the user must have the identical disk or CD loaded on the system for taking measurements every time the system needs to be identified.

Not all of these measurements need be made to develop an identification. Only a subset need be made. However, the measurements to be made must be determined prior to gathering the first identity and the identical measurements must be made every time the computer is to be identified.

The present invention employs a mix of publicly and privately available information that can be used to uniquely identify a computer system. The identification process of the present invention can be implemented in such a way that no duplications or falsifications are possible, making it useful for a keyless authentication scheme with the consequent reduction in key management costs and weaknesses.

The concept can be applied to scaled down (or minimal) devices and be used in copyright protection schemes. Also it can be extended up to identify and authenticate networks (Figure 4) of computers or to device copyright protection schemes for software.

Although the present invention has been discussed in the context of specific embodiments, one skilled in the art will realize that other measurement than those included in this discussion can serve to uniquely identify devices, systems and networks. The specific embodiments are the preferred embodiments but are not presented as limiting in any sense.

What is claimed is:

1. A method for identifying a computer system comprising the steps of:

    a. generating an authentication fingerprint of a first computer system comprising the steps of:

        i. gathering a first set of data comprising $n \geq 1$ timing sequences generated by at least one test that comprises measuring a circuit-level latency of at least one given operation by said at least one test being performed by said first computer system,

        ii. creating a first secure connection to an identification server from said first computer system,

        iii. sending said gathered first set of data to said identification server over said created first secure connection,

        iv. constructing an authentication fingerprint comprising a calculated statistical distribution of said $n \geq 1$ timing sequences of said sent first set of data, and

        v. storing in a storage media said authentication fingerprint at said identification server;

    b. testing a second computer system for identity with said first computer system comprising the steps of:

        i. gathering a second set of data as a verification sample comprising $m \geq 1$ timing sequences generated by said at least one test being performed by said second computer system,

ii. creating a second secure connection to said identification server from said second computer system,

iii. sending said verification sample to said identification server over said created second secure connection,

iv. comparing said $m \geq 1$ timing sequences of said verification sample with said authentication fingerprint to determine if said verification sample lies within said statistical distribution,

v. if step b.iv. succeeds, determining said first computer system and said second computer system to be identical, and

vi. if step b.iv. fails, determining said first computer system and said second computer systems to be not identical,

wherein, m and n can be equal or different

2. The method of claim 1, wherein said gathering steps a.i. and b.i. each further gather at least one other physical parameter of said first and second computer system, respectively, wherein said other physical parameter comprises temperature.

3. The method of any one of claims 1-2, wherein:

said statistical distribution of step a.iv. is generated by cluster analysis calculating a 2n-dimensional elliptic ball ($n \geq 1$) based on said first set of data gathered from said first computer system; and

said comparing step b.iv. determines if said verification sample lies within said elliptic ball.

4. The method of any one of claims 1-3 wherein said statistical distribution is modelled by linear regression.

5. The method of any one of claims 1-4 wherein said statistical distribution is modelled by a pattern-matching network.

6. The method of claim 5 wherein said pattern-matching network is a neural network.

7. The method of any one of claims 1-6 wherein, said statistical distribution is modelled by a combination of at least two statistical modelling techniques.

8. The method of claim 7 wherein said at least one of said at least two statistical modelling techniques is selected from the group consisting of cluster analysis and a pattern matching network.

9. The method of any one of claims 1-8 wherein, at least one of said first set of data, said second set of data, said authentication fingerprint and said verification sample is encrypted prior to any one of sending over a connection and storing in a storage medium.

10. The method of any one of claims 1-9 wherein at least one of said authentication fingerprint and said verification sample has been subjected to at least one of a minimum distance $k \geq 1$ coding scheme and secure hash function prior to any one of sending over a connection and storing in a storage medium.

11. The method of any of claims 1-10 wherein a zero-knowledge proof system is used on a least one sample in order to authenticate the computer system producing the verification samples to a given confidence level.

12. The method of any of claims 1-11 wherein said first and second computer systems each further comprises:

a. at least one CPU

b. at least one bank of memory having at least one portion;

c. at least one bus, said bus being shared by said at least one CPU and said at least one bank of memory; and

d. at least one clock sharing said at least one bus.

13. The method of any of claims 1-11 wherein, each of said first and second sets of data further comprises at least one other computer system identifier selected from the set consisting of addresses of hardware interfaces and system addressable hardware serial numbers, wherein the same computer system identifiers are contained in both said first and second sets of data.

14. The method of any of claims 1-11 wherein, each of said first and second sets of data further comprises at least one other computer system identifier selected from the set consisting of IP address of a device interface, hostname, user name, and combined serial and version numbers of application software and operating system, wherein the same computer system identifiers are contained in both said first and second sets of data.

15. The method of claim 12 wherein:

said at least one portion of said memory bank consists of one of CMOS memory and system RAM; and

said at least one test measures said circuit-level latency of said at least one portion of said memory bank.

16. The method of any of claims 1-15 wherein:

each of said first and said second systems further comprises a hard disk storage; and

said at least one test measures said circuit-level latency of said hard disk storage.

17. The method of any of claims 1-16 wherein:

each of said first and second computer systems further comprises at least one addressable internal device; and

said at least one test measures said circuit-level latency of said at least one addressable internal device.

18. The method of any of claims 1-17 wherein:

each of said first and said second computer systems further comprises at least one addressable external device; and

said at least one test measures said circuit-level latency of said at least one addressable external device.

19. The method of any one of claims 1-18 wherein:

each of said first and second computer systems comprises LANs having nodes that are physically connected by cables; and

said at least one test further comprises measuring the latency of network operations.

20. The method of any one of claims 1-19 wherein, each of said first and second computer systems comprises physically cabled devices.

21. The method of claim 20 wherein said physically cabled devices are selected from the group consisting of ATMs, point of sale terminals, and credit card validators.

22. The method of any of claims 1-21 wherein, each of said first and second computer systems comprises a wireless device connected to a server via a wireless LAN protocol.

23. The method of any of claims 1-22 wherein:

each of said first and second computers system comprises a handheld wireless device; and

said identification server comprises a base station.

24. The method of any of claims 1-23 wherein:

each of said first and second computer systems further comprises a biometric scanning device; and

said identification server comprises a database of both device fingerprints and the scanned biometric.

25. The use of the method of any of claims 1-24 in a digital signature scheme wherein the originating device is a part or full proxy for the signer.

26. The use of the method of any of claims 1-25 in combination with existing digital signature schemes.

27. The use of the method of any of claims 1-26 in an authentication scheme wherein, said first computer system is an originating device and is an entity to be authenticated.

28. The use of the method of any of claims 1-27 in an authentication scheme wherein, an originating device and a user are together considered as an entity to be authenticated.

29. The use of method of any of claims 1-28 in combination with existing authentication systems.

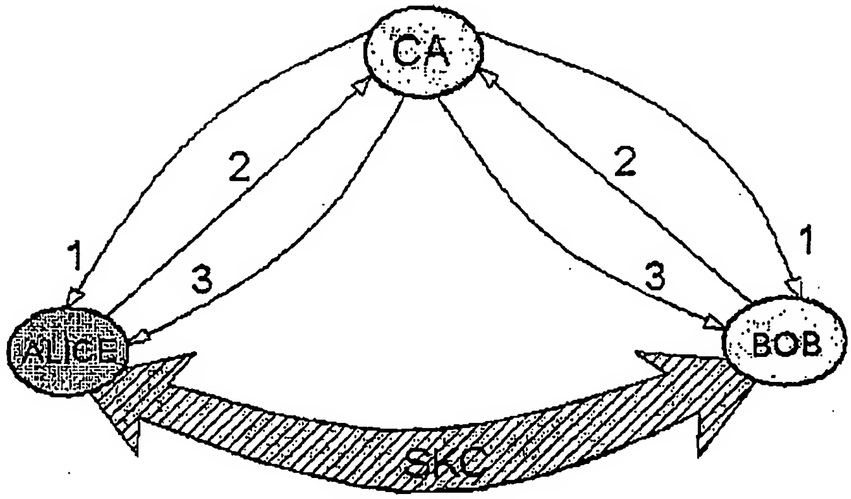

Figure 1

1. Alice and Bob contact the certificate authority (CA) to initiate the session.
2. CA request Alice and Bob to run Fingerprint Task and return a hashed version of their Finger Prints.
3. CA checks answers against his recorded values and sends Symmetric Keys (SK) to Alice and Bob.

Analysis of Data for Fingerprint

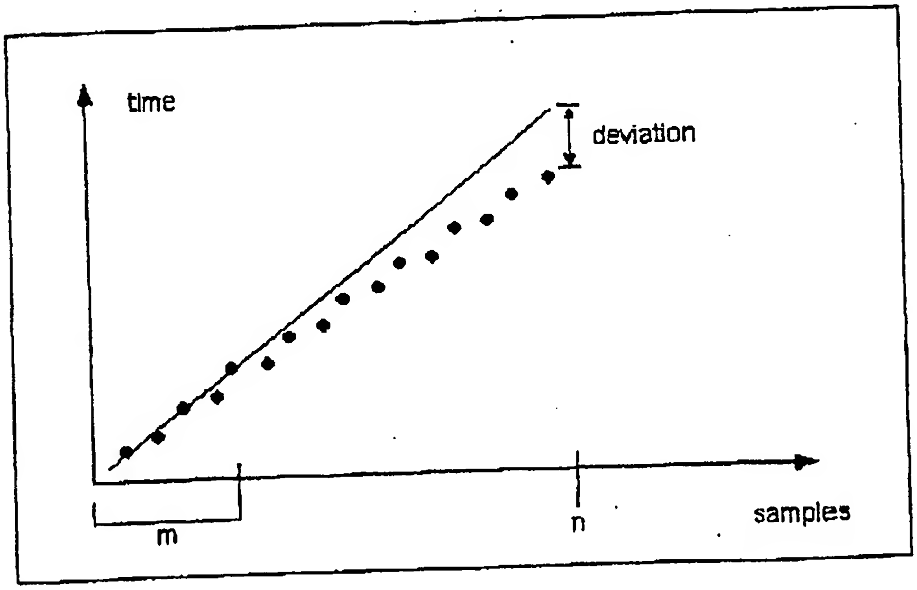

Figure 2

IE 0 2 0 4 2 9

- **CMOS-FP-1.3 Flow Chart**

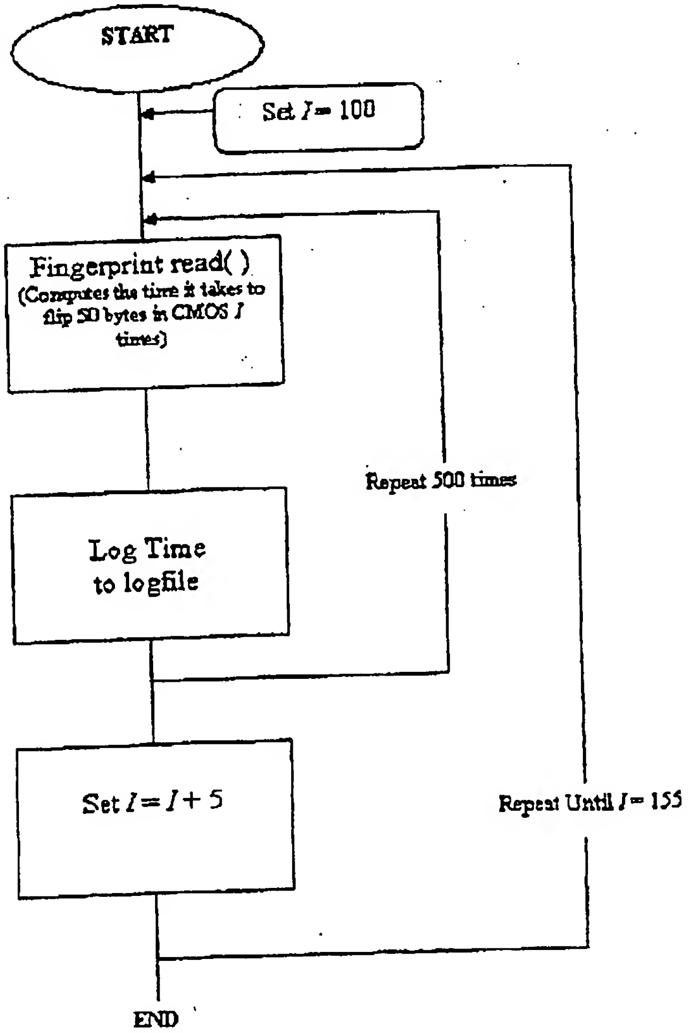

Figure 3

LAN Fingerprint

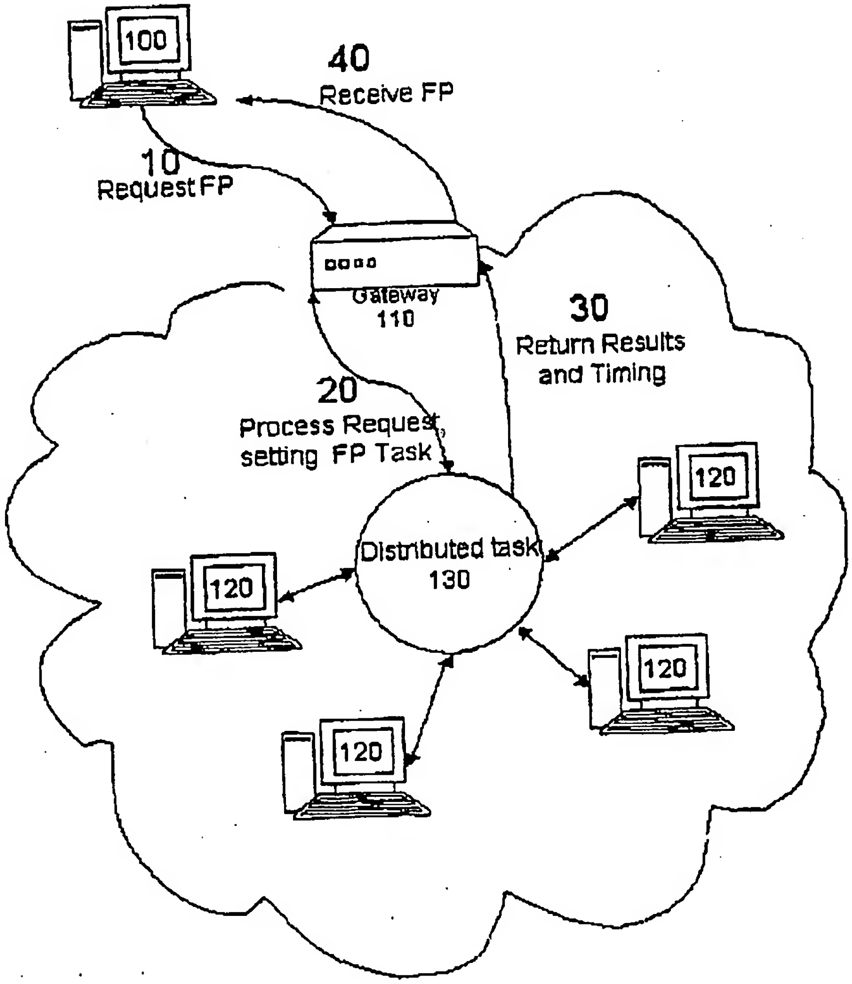

Figure 4